

COBOL-400

FOR THE COMPATIBLES/400 FAMILY OF COMPUTERS

GENERAL  ELECTRIC

COBOL-400

FOR THE COMPATIBLES/400 FAMILY OF COMPUTERS

COBOL-400, the Common Business Oriented Language, is available on General Electric's Compatibles/400 Information Processing systems. This universally-accepted business-oriented language is upward compatible through the entire GE-400 family of computers and gives you a solid, working partner for compatible hardware.

GENERAL ELECTRIC COBOL-400 FILLS SPECIFIC NEEDS

- The need for programming systems that can be processed on future, more powerful machines with a minimum of conversion costs in reprogramming and/or training. This is especially significant if you start modestly with a GE-415 and can foresee a possible expansion to a GE-435. Or, for that matter, any other large-scale General Electric Information Processor.
- The need for efficient translation of a programming system from one computer model to another of a different manufacturer.
- The need for a capability to meet the rapidly changing and expanding requirements of management which precludes constant revision and augmentation of programming systems. With COBOL-400, such changes and additions can be made with minimum time and costs.
- The need for a manner of producing extensive programming systems in a short period of time.

SPECIFIC ADVANTAGES

1. COBOL-400 programs are written in precise, easily learned English words and phrases. The language provides a clear method of expressing a problem, or "communicating" with a computer.
2. COBOL-400 programs can be run on another computer with minimum modification even though computer hardware characteristics are different.
3. COBOL-400 provides excellent documentation for problem definition and solution. Work started by one programmer and continued and/or completed by another is easily accomplished.
4. COBOL-400 simplifies the costly, time-consuming process of program testing. If necessary, it can be done efficiently by someone other than the original programmer.

5. COBOL-400 promotes use of standardized terminology among non-technical personnel and programmers thus inviting closer understanding of problems being solved.

6. COBOL-400 decreases training costs and significantly reduces retraining and reprogramming costs. Once a programmer is trained in COBOL techniques, he can change to another computer and use the same techniques.

HOW COBOL-400 WORKS

COBOL-400 is a communication vehicle. It is easy to learn and use. Briefly, here's how it works:

The programmer writes a COBOL-400 "source program" composed of English sentences and paragraphs, following the conventions of a standard reference format, to describe the data to be processed and to specify the required procedures. The source program is keypunched on cards which become input to the computer under control of the COBOL-400 compiler program (already loaded into the computer by the GE-400 Operating System). As output the COBOL-400 compiler produces an "object program" on either punched cards or magnetic tape. The object program is the actual sequence of machine instructions needed to accomplish the functions specified in the source program.

Additionally, the compiler produces an edited listing which includes a print-out of the English source program in the reference format. Another very important compiler function is to analyze the source program for clerical errors and to print error comments on any source program language errors it can detect.

The source program is sub-divided into four divisions specifying:

The identification of the program (IDENTIFICATION DIVISION)

The equipment to be used (ENVIRONMENT DIVISION)

The description of data to be processed (DATA DIVISION)

The sequence of procedures to be executed (PROCEDURE DIVISION)

For a complete, detailed description of each of these elements, see CPB-1001.

COBOL-400 options specify deviations from a well-defined “normal” mode of compilation. Options include:

- ## HARDWARE REQUIREMENTS

Hardware requirements differ but slightly between basic COBOL-400 and that required for COBOL-400 using the options. Configurations include:

Basic COBOL-400	COBOL-400 with options
ANY ONE OF THE COMPATIBLES/400 WITH	
8192 words of core memory	8192 words of core memory
(larger memory if desired)	(larger memory if desired)
4 magnetic tape units	5 magnetic tape units
Card Reader	Card Reader*
Card Punch	Card Punch*
Printer	Printer*
Console Typewriter	Console Typewriter
*(Magnetic tape unit may be substituted)	

COBOL-400 is based on COBOL-61 (the latest CODASYL specifications) which consist of:

REQUIRED COBOL — those elements which must be implemented by any company producing a COBOL compiler.

ELECTIVE COBOL — those elements which can be implemented at the manufacturer's discretion.

COBOL EXTENSIONS—elements of COBOL which have been added to the language since the publication of COBOL-61.

Detailed information is available at all Computer Department District Offices listed on back page, or call or write the Computer Department, Phoenix, Arizona.

GENERAL ELECTRIC

GE-400 COBOL

PROGRAMMER		JACK BRADBURY											
PROGRAM		EXAMPLE OF COBOL SOURCE LANGUAGE											
SEQ. NO.		A	B										
1	6	7	8	12	16	20	24	28	32	36	40		
100		IDENTIFICATION DIVISION.											
101		PROGRAM-ID.								INV-012			
102		AUTHOR.								JACK BR			
200		ENVIRONMENT DIVISION.											
301		CONFIGURATION SECTION.											
302		SOURCE-COMPUTER. GE-425.											
303		OBJECT-COMPUTER. GE-425 WITH SUPE											
304		CARD READER, CARD PUNCH, 8 MA											
305		INPUT-OUTPUT SECTION.											
306		FILE-CONTROL. SELECT FILE-ABC ASS											
307		ALTERNATE AR											
308		SELECT FILE-DEF ASS											
309	-	NATE AREA; P											
310		SELECT FILE-XYZ REN											
311		0407; RESERV											
300		DATA DIVISION.											
301		FILE SECTION.											
302		FD FILE-ABC, DATA RECORD IS STOCK											
303	-	ORDS, LABEL RECORD IS											
304		01 STOCK-RECORD.											
305		02 STOCK-NUMBER								PICT			
306		02 QTY-ON-ORDER								PICT			
307		02 UNIT-PRICE								PICT			
308		FD FILE-DEF, DATA RECORD IS TRANS											
309		OMITTED, RECORD CONTAINS											
310		01 TRANSACTION-CARD.											
400		PROCEDURE DIVISION.											
401		SET-UP. OPEN INPUT FILE-ABC, FILE											
402		MOVE ZEROS TO CONTROL-TOT											
403		INITIAL-READ. READ FILE-DEF, AT E											
404		START. READ FILE-ABC, AT END GO											
405		OF FILE-DEF IS GREATER THAN S											

Coding Explanation

LOCATION	PHOENIX , ARIZONA	PAGE	1	OF
	DATE	6/15/64		

44 48 52 56 60 64 68 72

3. ADBURY

RVISOR CONTROL, MEMORY SIZE 32K,
GENETIC TAPES.

IGN TO 0404, 0405; RESERVE 1
EA; BLOCKS ARE SERIAL-NUMBERED.
IGN TO 0100; RESERVE NO ALTER
PRIORITY IS 1.
NAMING FILE-ABC; ASSIGN TO 0406,
E 1; PRIORITY IS 2.

RECORD, BLOCK CONTAINS 20 REC
STANDARD.

URE IS X(12).
URE IS 9(8).
URE IS 99V99.
ACTION-CARD, LABEL RECORD IS
ENS 80 CHARACTERS.

DEF; OUTPUT FILE-XYZ.
ALS.
ND GO TO END-RTN-1.
TO END-RTN-3. IF STOCK-NUMBER
TOCK-NUMBER OF FILE-ABC, GO TO

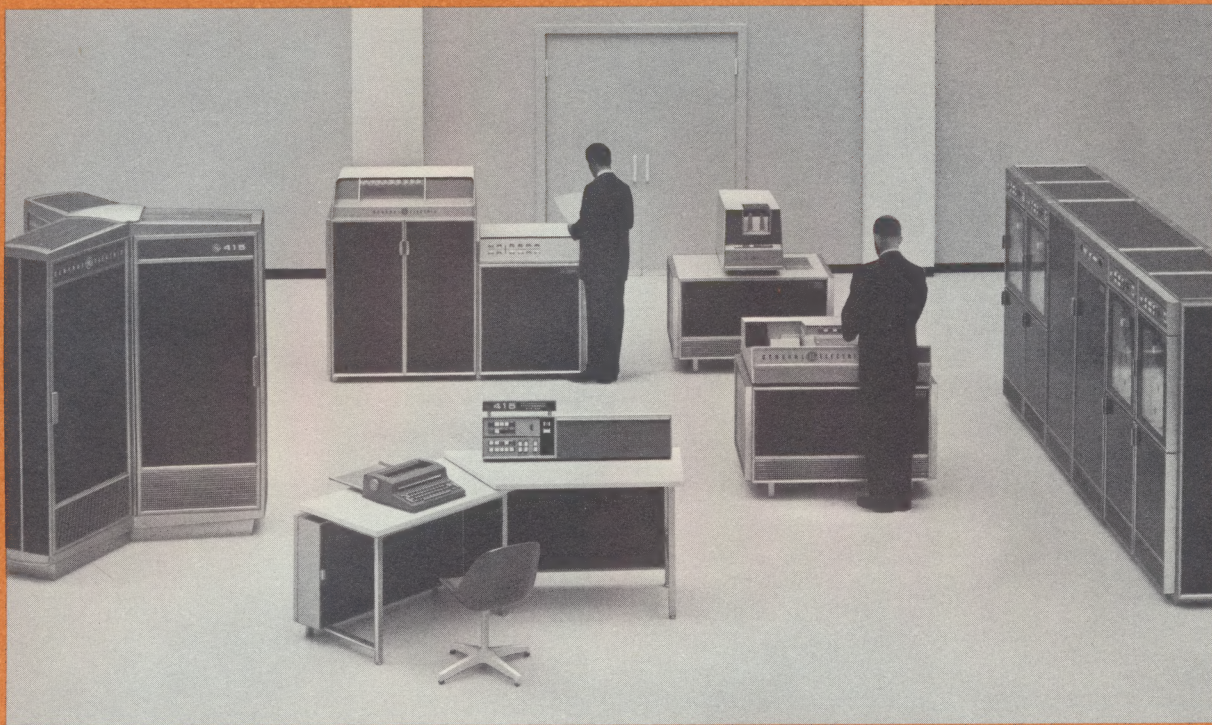
After compilation, the first 8 characters of "program name" will be used by the Program Monitor Routine as a search key for finding and loading the object program from a master instruction tape.

Compilation made on GE-425
Specification of computer that will use the object program.

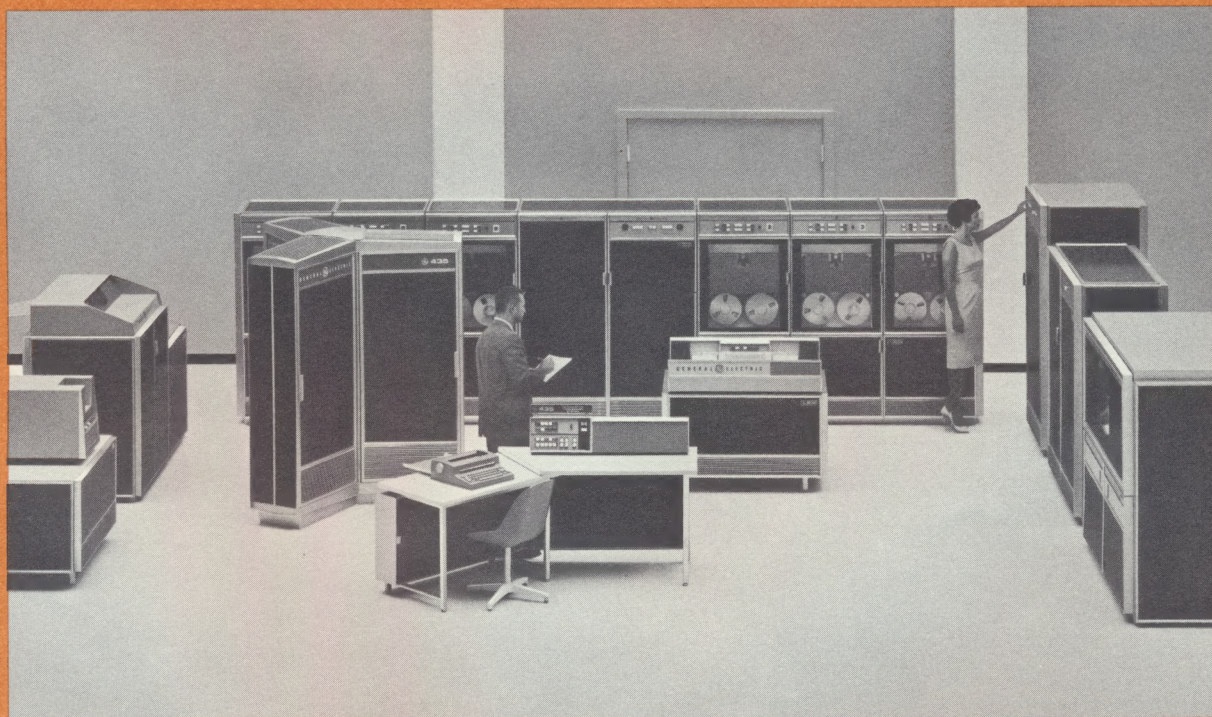
This paragraph contains one sentence for each file to be processed by the object program. The type of hardware device assigned to a file is indicated by the first two digits of the four octal digit device numbers following the word ASSIGN.

Assignment of record names
Defines how many integers in a record in the blocks.
Standard label records are to be used.

Typical imperative statement.
Typical conditional statement used in processing.
Directs the operations into the basic assembly program.



COBOL-400 IS AVAILABLE ON ALL GENERAL ELECTRIC COMPATIBLES/400 COMPUTERS, AND IS UPWARD COMPATIBLE THROUGH THE ENTIRE GE-400 FAMILY.



OFFICES

ATLANTA, GEORGIA
 BOSTON, MASSACHUSETTS
 CHARLOTTE, NORTH CAROLINA
 CHICAGO, ILLINOIS •
 CINCINNATI, OHIO
 CLEVELAND, OHIO •
 COLUMBUS, OHIO
 DALLAS, TEXAS •
 DAYTONA BEACH, FLORIDA
 DENVER, COLORADO
 DES MOINES, IOWA
 DETROIT, MICHIGAN
 HONOLULU, HAWAII
 HOUSTON, TEXAS
 HUNTSVILLE, ALABAMA
 INDIANAPOLIS, INDIANA
 JACKSONVILLE, FLORIDA
 KANSAS CITY, MISSOURI
 LOS ANGELES, CALIFORNIA

LOUISVILLE, KENTUCKY
 MEMPHIS, TENNESSEE
 MINNEAPOLIS, MINNESOTA
 NEW ORLEANS, LOUISIANA
 NEW YORK, NEW YORK •
 OKLAHOMA CITY, OKLAHOMA
 PHILADELPHIA, PENNSYLVANIA
 PHOENIX, ARIZONA •
 PITTSBURGH, PENNSYLVANIA
 PROVIDENCE, RHODE ISLAND
 SALT LAKE CITY, UTAH
 SAN FRANCISCO, CALIFORNIA
 SCHENECTADY, NEW YORK •
 SEATTLE, WASHINGTON
 ST. LOUIS, MISSOURI
 SYRACUSE, NEW YORK
 TALLAHASSEE, FLORIDA
 WASHINGTON, D. C. AREA •

Australia:

Australian General Electric Pty. Ltd.
 103 York Street, Sydney •
 552 Lonsdale Street, Melbourne •

Canada:

Canadian General Electric Co., Ltd.
 Electronic Equipment and Tube Dept.
 214 King Street West
 Toronto, Ontario, Canada

Europe:

International General Electric S.A.
 Data Automation Department
 1, rue du Temple
 Geneva, Switzerland

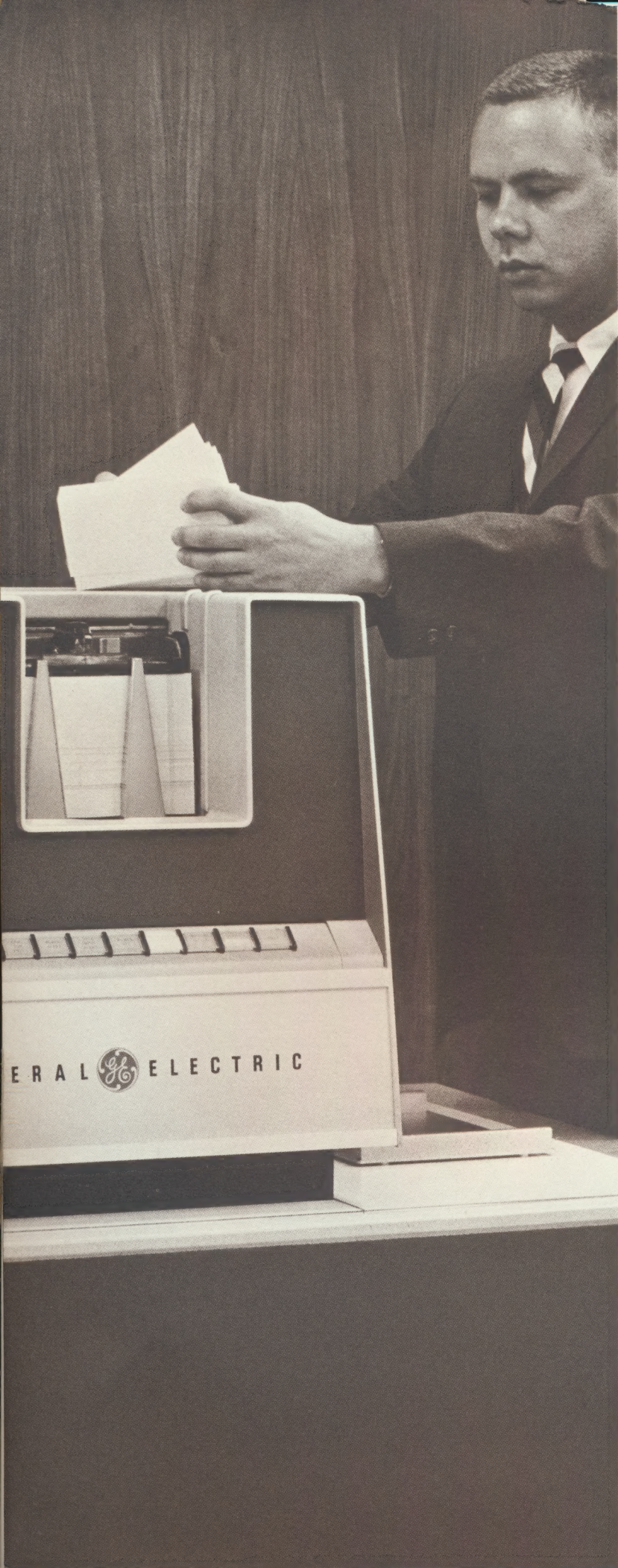
or write Drawer 270,
 Phoenix 1, Arizona

• Information Processing Centers
 in these cities offer complete
 computer services.

Progress Is Our Most Important Product

GENERAL  ELECTRIC

COMPUTER DEPARTMENT



EDP MANAGERS:

New demands on you for more and better EDP services require more than good hardware. You must have the right working relationship between hardware, programs, programmers, and operators. The proper software efficiently utilizes these four factors at minimum expense to meet demands for increased daily throughput. The Compatibles /400 Software System increases throughput on an hour-by-hour basis by:

- Providing an operating system to increase productivity by feeding more work through the computer in less time. The Operating System permits stacking of independent jobs for continuous operation.
- Providing a program segmentation capability that will minimize reassembly and recompile time in the program check out phase.
- Permitting simultaneous operation of unrelated programs to maximize the use of all peripheral equipment.
- Simplifying operator and programmer training through conformance to well established standards for programming and operating.
- Significantly reducing programming effort by providing all detailed input/output coding automatically.
- Providing a broad scope of the programming tools — tools that simplify the preparation of a wide variety of data processing and engineering applications.

The proper software system smooths out day-to-day computer operations and improves the ratio of cost to work done. The Compatibles/400 have such a system.

The Software System includes all the functions you need for normal business data processing.

- COBOL-61 Extended — for program standardization, compatibility, and thorough documentation.
- GE-400 FORTRAN — for your engineering or scientific problems.
- Macro Assembly Program — a compiler with assembly capabilities that is easy to use.
- Sort, Merge, and Report Program Generators — to produce fast, efficient programs for high volume production with greatly reduced programming effort.
- Standardized input/output routines that eliminate coding of I/O operations.
- Media conversion programs with simultaneous processing capabilities.
- Extensive debugging and testing aids — to assist in program checkout.

IT'S UPWARD COMPATIBLE

GE-400 software is upward compatible through the entire Compatibles/400 family. Programs you prepare for the GE-415 will work on the GE-425 and GE-435. Programs will run without alteration on the larger systems, and they'll

run faster. The GE-400 family has no options that hinder inter-computer compatibility.

IT'S QUALITY CONTROLLED

A user-oriented quality control group continually checks the programs and the write-ups, to see that manuals clearly describe how to use the software, and to see that the software performs as it is described in the manuals.

IT'S WELL DOCUMENTED

Simplified procedures, simple terms, and uncomplicated language let new programmers use the software quickly. Segmenting of programs permits testing smaller portions of the total job.

IT'S SIMILAR IN SOME RESPECTS

We didn't reinvent the wheel, but we removed the bumps to make the wheel roll smoother. Our automatic input/output system is like I/OCS. The Macro Assembly Program is like the AUTOCODER program.

IT'S A COMPLETE SYSTEM

It's a tape oriented, fully integrated system that takes advantage of all the powerful hardware features. There are no central processor options to complicate software requirements.

GE-400 COBOL BENEFITS

Many detailed programming requirements are eliminated when using COBOL. The COBOL language provides a clear method of expressing a problem to a computer. COBOL also provides excellent documentation for the problem definition and method of solution. Concise documentation permits work begun by one programmer to be continued and finished by another programmer. Source programs written for one computer can be used with little or no alteration on another computer — even if equipment characteristics are different.

The use of the COBOL language can decrease training costs. Once a programmer is trained in COBOL techniques, he can transfer to another computer and use the same techniques. GE-400 COBOL compiles all the required COBOL-61 Extended and selected elements of Elective COBOL. If your programming staff is familiar with COBOL techniques, they can code programs for the Compatibles/400. Programs are upward compatible through the GE-400 family of computers.

GE-400 FORTRAN

This compiler greatly reduces detailed program preparation time for scientific and engineering problems. Scientists and engineers can run their problems written in FORTRAN language on the Compatibles/400.

The computer compiles a series of machine instructions for the FORTRAN language statements. The GE-400 FORTRAN compiler translates statements into machine language — and the scientist or engineer doesn't have to worry about details of machine language coding.

Using the GE-400 FORTRAN compiler will reduce the possibility of clerical errors. Changes and modifications to a problem written in GE-400 FORTRAN language are easier than changes to that same problem if it were done in machine language. Typical FORTRAN statements compiled by the GE-400 FORTRAN compiler includes:

GENERAL ELECTRIC		GE-400 COBOL	
PROGRAMMER		JACK BRADBURY	
PROGRAM		EXAMPLE OF COBOL SOURCE LANGUAGE	
SEQ. NO.	A	B	
100	IDENTIFICATION DIVISION.		
101	PROGRAM-ID.		INV-0123.
102	AUTHOR.		JACK BRADBURY
200	ENVIRONMENT DIVISION.		
201	CONFIGURATION SECTION.		
202	SOURCE-COMPUTER.		GE-425.
203	OBJECT-COMPUTER.		GE-425 WITH SUPERVISOR
204	CARD READER, CARD PUNCH, & MAGNETIC		
205	INPUT-OUTPUT SECTION.		
206	FILE-CONTROL. SELECT FILE-ABC ASSIGN TO		
207			ALTERNATE AREA; BU
208			SELECT FILE-DEF ASSIGN TO
209			NATE AREA; PRIORI
210			SELECT FILE-XYZ RENAMING
211			0407; RESERVE 1; P
300	DATA DIVISION.		
301	FILE SECTION.		
302	FD FILE-ABC, DATA RECORD IS STOCK-RECO		
303			ORDS, LABEL RECORD IS STAND
304	01 STOCK-RECORD.		
305	02 STOCK-NUMBER		PICTURE IS
306	02 QTY-ON-ORDER		PICTURE IS
307	02 UNIT-PRICE		PICTURE IS
308	FD FILE-DEF, DATA RECORD IS TRANSACTION		
309			OMITTED, RECORD CONTAINS 80
310	01 TRANSACTION-CARD.		
400	PROCEDURE DIVISION.		
401	SET-UP. OPEN INPUT FILE-ABC, FILE-DEF;		
402			MOVE ZEROS TO CONTROL-TOTALS.
403	INITIAL-READ. READ FILE-DEF, AT END GO		
404	START. READ FILE-ABC, AT END GO TO END		
405			OF FILE-DEF IS GREATER THAN STOCK-

GENERAL ELECTRIC		GE-400 FORTRAN	
	READ(3,100)A,B,C,D,(E(I),I=1,1,10)		
100	FORMAT (4F10.4/10E6.2)		
	WRITE (5,200)X(2,3,2),Y,Z		
200	FORMAT (F 20.8,2F15.6)		
	$X = (-B + \sqrt{B^2 - 4.0 * A * C}) / (2.0 * A)$		

Coding Explanation

PHOENIX, ARIZONA PAGE OF
DATE 6/15/64

After compilation, the first 8 characters of the program identification will be used by the Program Monitor Routine as a search key for finding and loading the object program from a master instruction tape.

Compilation made on GE-425

Specification of computer that will use the object program.

This paragraph contains one sentence for each file to be processed by the object program. The type of hardware device assigned to a file is indicated by the first two digits of the four octal digit device numbers following the word ASSIGN.

Standard label records are to be used.

Assignment of data names, and definition of the characteristics of fields, records, and files to be processed by this program.

Typical imperative statement.

Typical conditional statement used in processing.

CONTROL, MEMORY SIZE 32K,
TAPES.

0404, 0405: RESERVE 1
BLOCKS ARE SERIAL-NUMBERED.

0100: RESERVE NO ALTER
Y IS 1.

FILE-ABC: ASSIGN TO 0406,
PRIORITY IS 2.

D, BLOCK CONTAINS 20 REC
ARD.

X(12).

9(8).

99V99.

CARD, LABEL RECORD IS
CHARACTERS.

OUTPUT FILE-XYZ.

TO END-RTN-1.

RTN-3. IF STOCK-NUMBER

NUMBER OF FILE-ABC, GO TO

Coding Explanation

Read from device 3, under format control
specified in statement 100.

This is the 100 format statement.

Write on device 5 under control of
format specified in statement 200.

Write X from a three dimensional array,
and Y and Z.

This is the 200 format statement.

$$X = \frac{-B + \sqrt{B^2 - 4AC}}{2A}$$

MACRO ASSEMBLY PROGRAM

It's more than a one-for-one assembly program — it's a compiler with assembler capabilities. It saves man hours because the program generates detailed instructions that previously had to be written individually. As in COBOL, the Macro Assembly Program has four divisions: Identification, Environment, Data, and Procedure. Programmer learning time is decreased because of this similarity with COBOL.

In the Macro Assembly Program you describe each data file by giving a data name to each logical field contained in the file. These fields (portions of records) are handled as units. Programmers familiar with AUTOCODER programming techniques will find the Macro Assembly Program Procedure Division easy to use. In addition to macro instructions, all basic machine instructions are available in mnemonic form. These basic mnemonics can be intermixed with macro instructions.

The output listing of a compilation provides a complete record of any detected program errors to aid in debugging. The output listing can also include a symbol analysis, i.e., a cross reference list showing what instructions modify or refer to other instructions and data in a given program. This listing gives page and line numbers for each referenced symbol.

GENERAL ELECTRIC

Compatibles/400 Programming Form

PROGRAM										EXAMPLE OF MACRO ASSEMBLY										PROGRAMMER A. E.													
SEQUENCE		TYPE		REFERENCE SYMBOL										OPERATION																			
				DATA NAME										LEVEL										SYMBOL									
0010				IDENTIFICATION DIVISION																													
0020				SEGMENT										PAY001																			
0030				AUTHOR										A. E. MERNER																			
0040				TITLE										SAMPLE PAYROLL																			
0050				IDEN										PAY																			
0060				ENVIRONMENT DIVISION																													
0070				SRMETHD										CALL																			
0080				DEFCTL										INT																			
0090				DEFOPEN										OPEN1																			
0100				DEFCLQS										CLOSE1																			
0110				DEFEOR										EOR1																			
0120				DEFCKPT										PAUSE; DUMP																			
0125				DATA DIVISION																													
0130				FILE SECTION																													
0140				PAYMSTR										FD																			
0150				DEVICE										09.03																			
0160				BLKSIZE										3																			
0170				IOMETHD										EXCH																			
0180				EOFADDR										END1																			
0190				LABDEF										STND																			
0975				PROCEDURE DIVISION																													
0980				START										LAL										0									
0990				OPEN										PAYMSTR; TIMECLS																			
1000				READM										READ										PAYMSTR									
1010				READTC										IF										SW1 ON GO TO TEST									
1020				RDTC										READ										TIMECLS									
1030				TEST										COMPARE										PAYNO; TC PAYNO; TEST									
1040														MOVE										TC PAYNO; ERMSG2A									
1050														TYPE										OUT; ALP; ERMSG2									

Coding Explanation

PAY 001 is the name of this segment.

Author's name printed on top of each assembly output page.

Program name printed on each page.

Identification code "PAY" will be punched into output cards.

Specifies method of obtaining subroutines to be used in program.

Defines type of buffer processing & scheduling routines to be generated.

Specifies type of file OPEN routine to be generated.

Specifies type of file CLOSE routine to be generated.

Generates end of reel processing routine.

Specifies checkpoint. Provides parameters for ckpt routine.

This is the data name of the file to be described.

Logical device code of the device that contains the file.

Blocking factor for tape or disc files.

Generates coding for use of buffer areas in memory.

Provides address of user-coded routine to be used at EOF.

Specifies type of label processing to be used for this file.

Basic Assembler Language mnemonic — sets accum. to locs. 0-3.

Open the paymaster and time card files.

Read a record from the paymaster file.

If switch 1 is ON, go to routine label TEST.

Read one record from time card file.

Compare the time card record with the master record.

Move time card number to error message in memory.

Type out error message.

FEATURES

Programmers will recognize the benefits of these features :

- Arithmetic macros such as add, subtract, multiply and divide, including options on each macro to round or truncate the result.
- Control macros such as Compare, If, Set Switch, Go To, Go To Depending On, Halt, Abort.
- Data movement macros such as Move, Load, Unload.
- Editing features include automatic zero suppress, float dollar sign, asterisk protect, and others.
- Sequence checking of source input programs.
- Can call segments from a library tape and have them automatically loaded by the loader at execution time.
- Complete File processing macro instructions. The Macro Assembly Program makes it easy for even the relatively inexperienced programmer to code complete applications. To get a record from a file, the programmer only needs to write the word READ. To write a record, the programmer only needs to write the one instruction WRITE. Coding for all intermediate steps is supplied by the Macro Assembly Program. This procedure is similar to GET and PUT in AUTOCODER programs.

TAILORED SORTING AND MERGING PROGRAMS WITH THE SORT AND MERGE PROGRAM GENERATORS

The Sort and Merge Program Generators produce fast efficient object programs with minimum effort. The generated programs use the very powerful tournament technique and considerably reduce times required to sort and merge large files. The generators have options for insertion of your own coding for manipulation of records or files prior to, during, and following the sort. The sorts and merges are tailored at generation time, according to the descriptive parameters provided by the programmer. Extensive options give flexibility in key selection. The Sort and Merge Programs use the Scatter/Gather capability built into the hardware, and the Dual Controller feature of the magnetic tape controller. Here's how easy it is to use.

GENERAL ELECTRIC															Compatibles/400 Programming Form																						
PROGRAM															PROGRAMMER																						
SORT GENERATOR - EXAMPLE																																					
SEQUENCE		TYPE		REFERENCE SYMBOL										OPERATION										PICTURE		OCCURS											
				DATA NAME										LEVEL										C													
1		5		16 17 18 19 20 21 22 23 24 25																				40 41													
				IDENTIFICATION DIVISION																																	
				SEGMENT										SORTA																							
				AUTHOR										OPAL WILSON																							
				TITLE										SORT PERSONNEL																							
				IDEN										SORT																							
				ENVIRONMENT DIVISION																																	
				MEMSIZE										6000																							
				DEFCKPT										PROCEED; DUMP																							
				WRKTAP										403; 404; 405; 406																							
				DATA DIVISION																																	
				FILE SECTION																																	
				PERSIN										FD																							
				DEVICE										402																							
				FILETYP										IN																							
				BLKSIZE										20																							
				LABDEF										STND																							
				LABOPT										1100																							
				LABCON										0005PERSIN																							
				PERREC										01																							
														02																							
				NAME										02																							
														02																							
				PERSOUT										FD																							
				FILETYP										OUT																							
				LABDEF										STND																							
				LABOPT										1110																							
				LABCON										0000PERSOUT																							
				PROCEDURE DIVISION																																	
				SORT										USING																							
														NAME																							
														END																							

Coding Explanation

Name of this segment of this sort problem is Sort A.

Program name to be printed on each page of printer output.

Amount of memory available to sort.

Provides normal restart capabilities (checkpoint).

There are working tapes mounted on handlers, 3,4,5,6.

Personnel file is file description.

Personnel file is on handler 2.

The personnel file is an input file.

The block size is 20 records per block.

The tapes have standard labels.

A definition of the label options.

Specific creation date to be checked by formula.

Name of record is personnel record.

There is a 10 character field within the personnel record.

There are 16 alphabetic characters in this field.

There are 14 numeric characters in this field.

Defines the output file.

It is an output file.

Use standard labels.

A definition of the label options.

Specific label information to be written on the output tape.

Sort the file in alphabetic order on the names.

Report preparation is easy because this efficient generator eliminates the tedious coding often associated with report writing. The Identification, Environment, and Data divi-

sions are exactly the same as those used in the Macro Assembly Program. The Procedure division is replaced by the Report Description Form, defining the report. You can insert your own coding to tailor the object program to your unique requirements.

GENERAL ELECTRIC										Compatibles/400 Programming Form										Report Description Form																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																		
PROGRAM															PROGRAMMER																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
EXAMPLE OF REPORT PROGRAM GENERATOR															MICHAEL																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67

GILTNER

[illegible]

DB AMOUNT

59

99- @Z, ZZZ, ZZZ 99-
99- @Z, ZZZ, ZZZ 99-
99- @Z, ZZZ, ZZZ 99-
99- @Z, ZZZ, ZZZ 99- @Z, ZZZ, ZZZ 99-

Describes point where user's own coding is called in for calculating cross footing totals.

ELIMINATE DETAILED CODING WITH THE BASIC INPUT/OUTPUT SUPERVISOR

The Input/Output Supervisor is one software element that is used by all types of programs. The functions of the supervisor are standard, and are available to all programs. This is beneficial because it reduces the time required to assemble individual programs. Basic hardware-oriented instructions need not be assembled with each program.

You don't need to store coding for peripherals you don't have in your system. This is a partial list of the functions provided by the Basic Input/Output Supervisor.

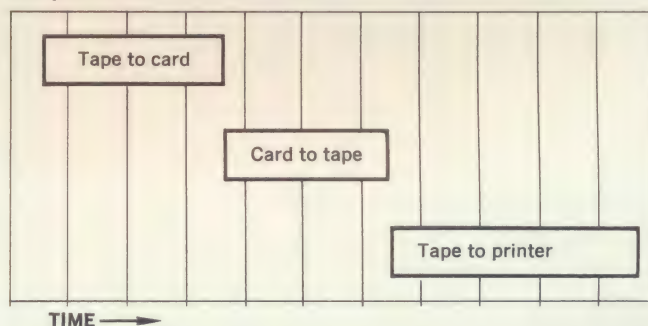
- Card read and punch control
- High speed printing control
- Disc Storage Unit Control (optional)
 - Seeking/reading/writing control
 - Key analysis
 - Random access technique selection
- Perforated tape reading/punching control (optional)
- MICR document reading control (optional)
- Typewriter control
- Processor channel interrupt control
- Magnetic Tape controls
 - Dual channel usage control
 - Execution of basic input/output commands
 - Checkpoint writing
- Control of simultaneous read/write/compute
- Standard procedures for equipment error correction or recovery
- Read and write error detection and recovery
- Input/output error detection

PROGRAMS FOR MULTI-PERIPHERAL OPERATIONS

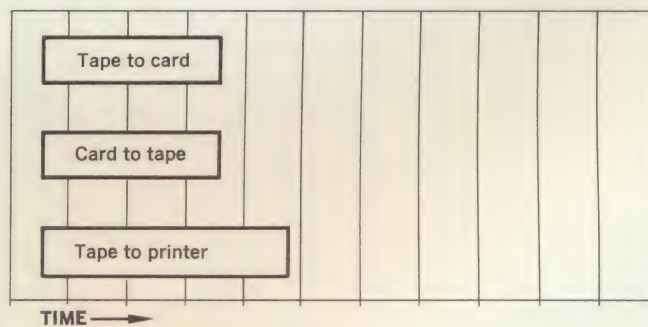
The Software System utilizes hardware features to do conversion work in a very efficient manner. With multiple input/output channels available, the Compatibles/400 software can run peripheral operation programs simultaneously. The software to accomplish this is supplied by General Electric.

MULTI-WAY MEDIA CONVERSION

There is a substantial time saving involved when several programs are run at one time. In some cases in the past, the jobs were done like this:

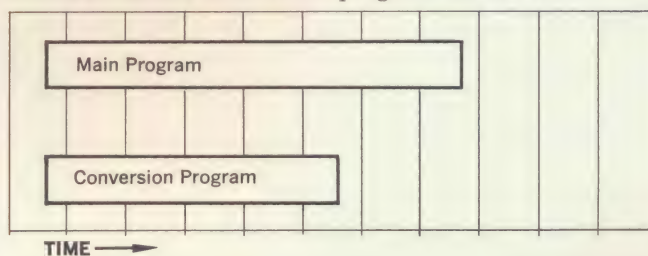


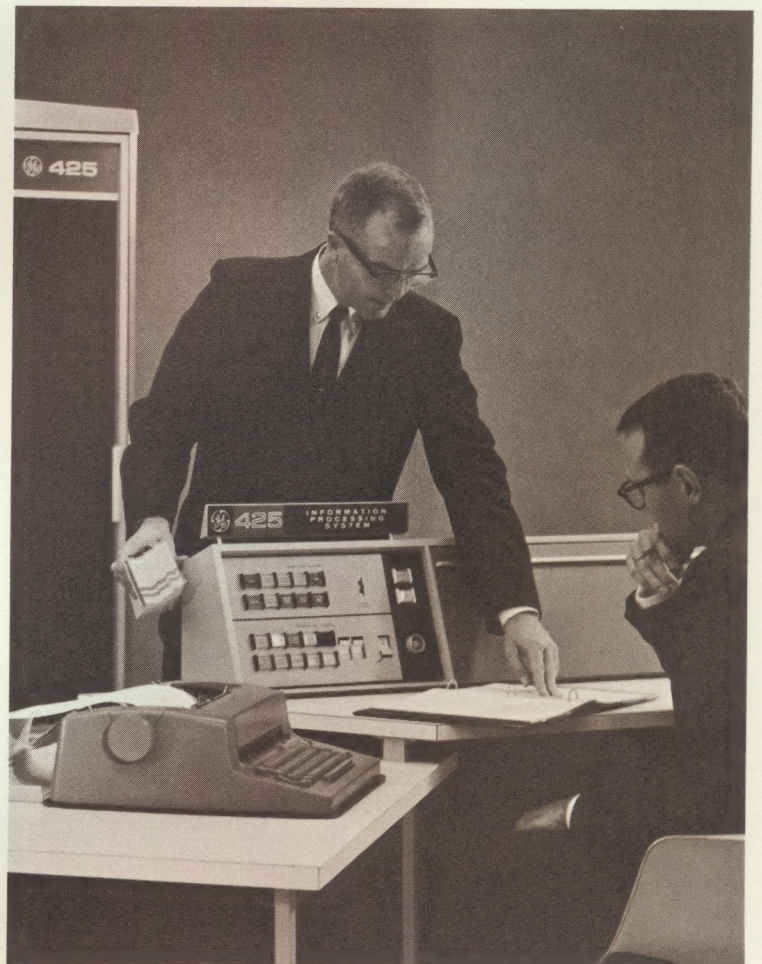
The Multi-Way Media Conversion programs with the Compatibles/400 let you do the jobs simultaneously like this:



MEDIA CONVERSION AND MAIN PROGRAM

An unrelated main program can be run simultaneously with one of the Media Conversion programs like this:





MORE WORK FOR YOUR DOLLAR WITH THE GE-400 OPERATING SYSTEM

Many computer users scrutinize microsecond speeds within the processor, but overlook the elapsed time between jobs. The GE-400 Operating System reduces the elapsed time between jobs by maintaining the computer in continuous operation during the processing of a group of independent jobs.

The GE-400 Operating System increases the ratio of useful computer time to total available computer time. For example, it reduces the amount of job set-up time by permitting stacking of independent unrelated jobs in the card reader, or on tapes. The system does not depend on, or wait for a decision to be made by the operator.

The GE-400 Operating System includes the following functions:

- Beginning and end-of-job logging. This gives a hard copy record of all jobs started and completed.
- GE-400 Loader. Loads output from any GE-400 language processor. It loads relocatable segments that were written separately.
- Sequential or controlled job-to-job processing.
- Load and go. Permits testing of a newly written program immediately after it has been assembled. You can assemble a program and give it a test run all within the same computer run.
- Debugging links. Provides an easy way of searching for and transferring control to a debugging aid such as a memory dump.
- Checkpoint recovery. Uses the information from a selected checkpoint to perform the following functions:
 - Repositioning of files on peripheral units
 - Restoration of the object program's memory.

HELPFUL, TIME-SAVING AIDS FOR YOUR PROGRAMMERS

Special aids to programmers get your programs on the air sooner by providing a full range of debugging aids and efficient means for including the aids in a tape operating environment. Routines are available for reducing test preparation time. One such routine is the Test Data Dispersion Routine which creates a variety of data for testing programs during the debugging stage. The routine also generates random sequence records for checkout of sorting and merging programs.

The Librarian creates and maintains the system tape, library tape, and production program tapes.

Other routines provide for maintenance of source programs on magnetic tapes. This reduces card storage requirements and decreases source program updating time.

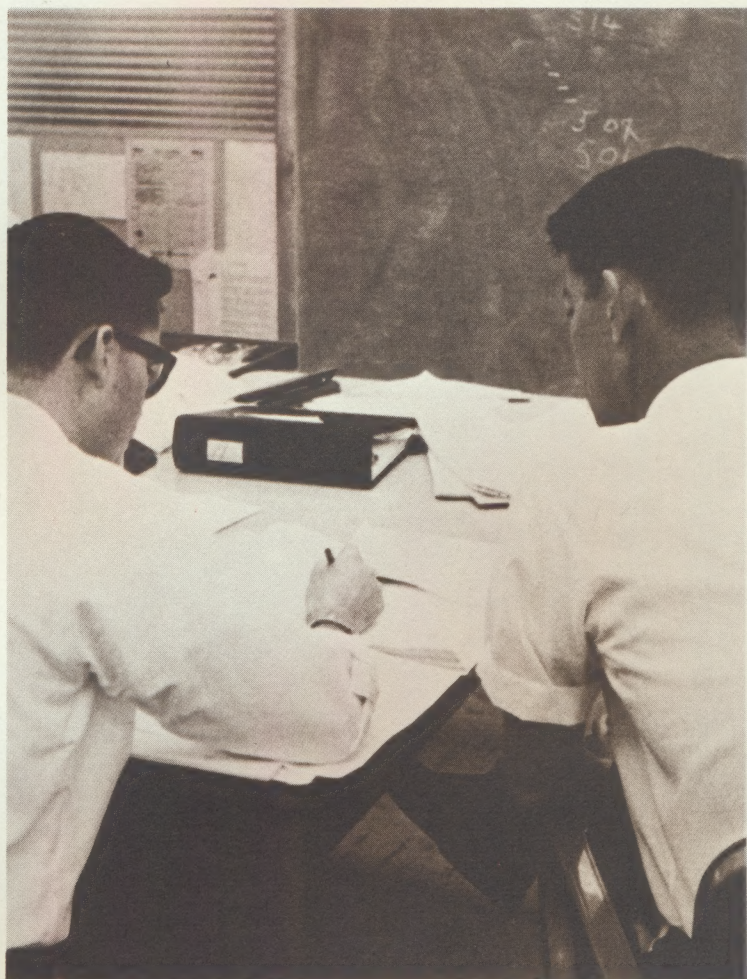
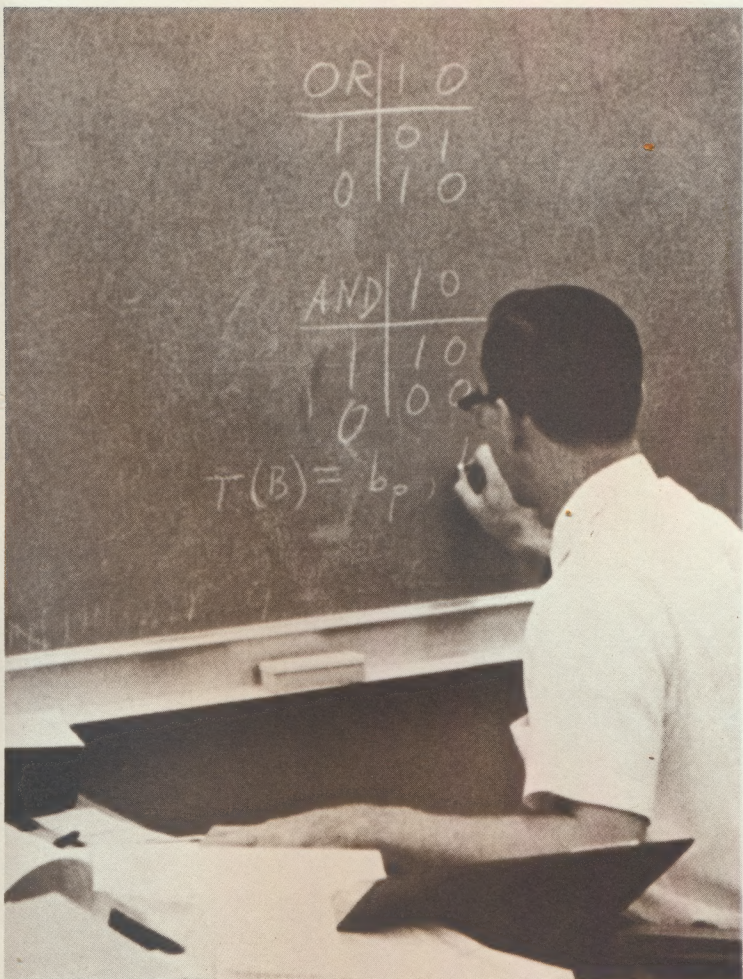
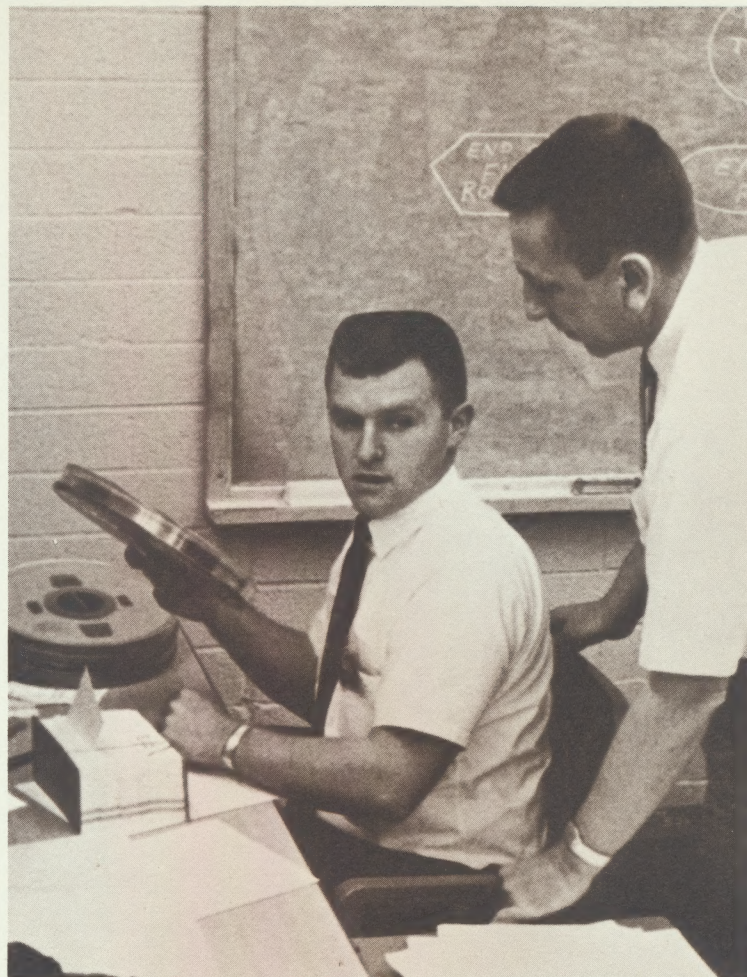
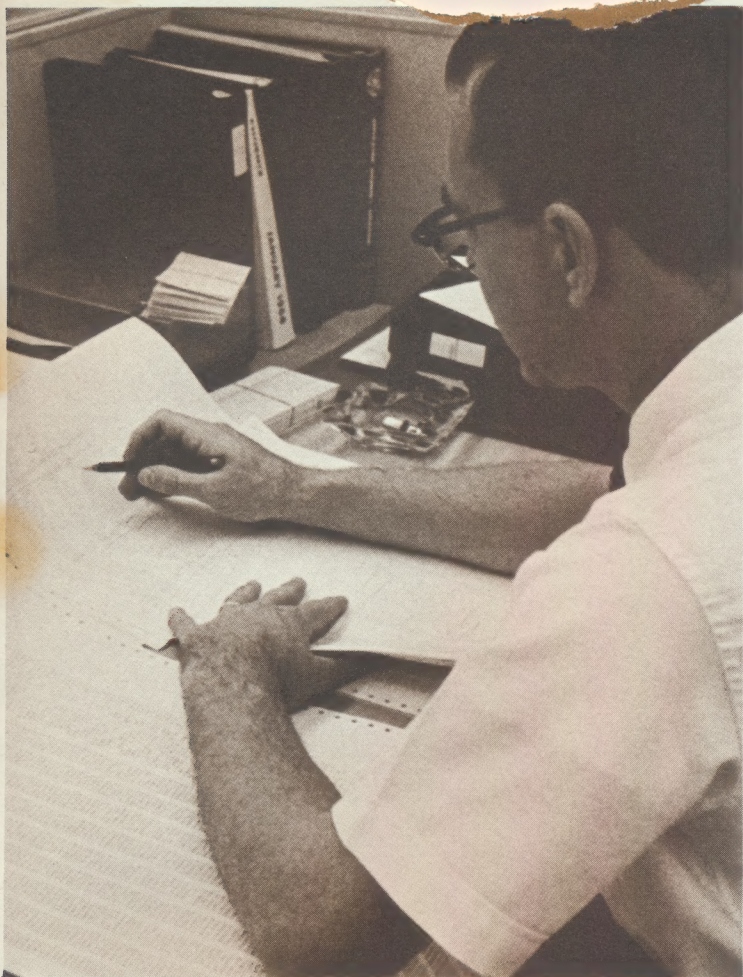
SUMMARY

The Software System that incorporates language processors, an operating system, automatic I/O coding, and simultaneous operations can be yours with the Compatibles/400 system.

These time and money saving advantages are available with all members of the Compatible/400 family.

Compare the software features that account for the high performance of the Compatibles/400 with those of any other system in their price range. Program compatibility within the family means you can amortize your programming costs over a longer period. Programs you prepare for the GE-415 operate without change on the GE-425, or GE-435. If you've outgrown your 1401, the 1401 Compatibility Option lets you make a smooth, easy changeover to the Compatibles/400.

Call or write the nearest Computer Department office of the General Electric Company for full details on the Compatibles/400.



OFFICES

ATLANTA, GEORGIA
 BOSTON, MASSACHUSETTS
 CHARLOTTE, NORTH CAROLINA
 CHICAGO, ILLINOIS •
 CINCINNATI, OHIO
 CLEVELAND, OHIO •
 COLUMBUS, OHIO
 DALLAS, TEXAS •
 DAYTONA BEACH, FLORIDA
 DENVER, COLORADO
 DES MOINES, IOWA
 DETROIT, MICHIGAN
 HONOLULU, HAWAII
 HOUSTON, TEXAS
 HUNTSVILLE, ALABAMA
 INDIANAPOLIS, INDIANA
 JACKSONVILLE, FLORIDA
 KANSAS CITY, MISSOURI
 LOS ANGELES, CALIFORNIA

LOUISVILLE, KENTUCKY
 MEMPHIS, TENNESSEE
 MINNEAPOLIS, MINNESOTA
 NEW ORLEANS, LOUISIANA
 NEW YORK, NEW YORK •
 OKLAHOMA CITY, OKLAHOMA
 PHILADELPHIA, PENNSYLVANIA
 PHOENIX, ARIZONA •
 PITTSBURGH, PENNSYLVANIA
 PROVIDENCE, RHODE ISLAND
 SALT LAKE CITY, UTAH
 SAN FRANCISCO, CALIFORNIA
 SCHENECTADY, NEW YORK •
 SEATTLE, WASHINGTON
 ST. LOUIS, MISSOURI
 SYRACUSE, NEW YORK
 TALLAHASSEE, FLORIDA
 WASHINGTON, D.C. AREA •

Australia:

Australian General Electric Pty. Ltd.
 103 York Street, Sydney •
 552 Lonsdale Street, Melbourne •

Canada:

Canadian General Electric Co., Ltd.
 Electronic Equipment and Tube Dept.
 214 King Street West
 Toronto, Ontario, Canada

Europe:

International General Electric S.A.
 Data Automation Department
 1, rue du Temple
 Geneva, Switzerland

or write Drawer 270,
 Phoenix 1, Arizona

• Information Processing Centers
 in these cities offer complete
 computer services.

Progress Is Our Most Important Product

GENERAL  ELECTRIC

COMPUTER DEPARTMENT